This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2766844, IEEE Access

A LogitBoost-based Algorithm for Detecting Known and Unknown Web Attacks

Muhammad Hilmi Kamarudin, *Member, IEEE*, Carsten Maple, Tim Watson, and Nader Sohrabi Safa

***Abstract*—The rapid growth in the volume and importance of web communication throughout the Internet has heightened the need for better security protection. Security experts, when protecting systems, maintain a database featuring signatures of a large number of attacks to assist with attack detection. However, used in isolation, this can limit the capability of the system as it is only able to recognise known attacks. To overcome the problem, we propose an anomaly based intrusion detection system using an ensemble classification approach to detect unknown attacks on web servers. The process involves removing irrelevant and redundant features utilising a filter and wrapper selection procedure. Logitboost (LB) is then employed together with Random Forests (RF) as a weak classifier. The proposed ensemble technique was evaluated with some artificial datasets namely NSL-KDD, an improved version of the old KDD Cup from 1999, and the recently published UNSW-NB15 dataset. The experimental results show that our approach demonstrates superiority, in terms of accuracy and detection rate over the traditional approaches, whilst preserving low false rejection rates.**

***Index Terms*—Anomaly detection, intrusion detection, data mining, classification, web attacks**

I. Introduction

The increasing number and frequency use of web-based applications and web servers have resulted in a greater necessity for effective security defence in both in home and enterprise networks. Many organisations realise the urgency of utilising security protection tools to preserve their computer servers and reduce the impact of catastrophic attacks. A comprehensive analysis carried out by Symantec [1] reveals that nearly one million new threats are released into the public network each day. The recent attack on 21st October 2016 [2] was specifically designed to target Dyn a major Internet infrastructure company. The attack is recognised as one of the largest attacks with millions of source IP addresses used to request DNS lookup. Dyn is responsible for providing DNS service translations, that is translating human-friendly site names into machine-readable Internet addresses. The attack nearly brought down the entire US Internet service. Vulnerable Internet of Things (IoT) devices such as webcams and digital videos can be used to distribute malicious software and spam. *Mirai* is an example of software that was designed to exploit the vulnerabilities in IoT devices by infecting them. The infected devices were turned into slave or zombie devices and formed an army of bots that was used to perform large scale Distributed Denial of Services (DDOS) attacks from multiple different locations. The attacks caused outages and slowness for many of Dyn's customers including Twitter, Paypal, CNN, and some businesses hosted by Amazon.com Inc.

A more recent massive cyber-attack took place on 12th May 2017 and major impact in a significant element of the UK's National Heath Service (NHS), other health industries and created chaos in hospitals across England. Thousands of computers at hospitals and GPs surgeries became victims of global ransomware attacks, derivatives of the *WannaCry* attack, which are believed to have exploited a vulnerability first discovered by the National Security Agency (NSA) [3]. In particular, the attack exploited a vulnerability in the Windows Server Message Block (SMB) protocol and installed backdoor tools to deliver and run a WannaCry ransomware package.

Although the Internet provides convenient real-time information services to the public, the potential threats to confidentiality, integrity and availability (CIA) need to be addressed more effectively and permanently [4]. To fortify the security aspects of web-based servers and systems, Intrusion Detection Systems (IDSs) can be used as a complimentary device to many existing security appliances such as password authentication, firewalls, access control and vulnerability assessments.

An IDS is an application system or device that functions to identify either hostile activities or policy violation activities within a network. IDSs play an active role in network surveillance, as well as functioning as a network security guard and have been widely used in recent years as a network security component. They are employed to capture and analyse traffic movement and send an alarm when intrusive actions are detected. The alarm alerts the security analyst, who then takes necessary action. In general, IDSs can be classified as either a network-based IDS (NIDS) or as a host-based IDS (HIDS) to recognise signs of intrusion [5]. The classification is based on whether the placement of the IDS is intended either to capture traffic for the whole network or only for a

M.H. Kamarudin is with the Cyber Security Centre, WMG, University of Warwick, Coventry, United Kingdom (e-mail: hilmi_kamarudin@yahoo.com)

C. Maple now is the Director of Research at the Cyber Security Centre, WMG, University of Warwick, Coventry, United Kingdom (e-mail: cm@warwick.ac.uk).

T. Watson is the Director of the Cyber Security Centre at WMG, University of Warwick, Coventry, United Kingdom (e-mail: tw@warwick.ac.uk).

N. S. Safa is the Researcher of the Cyber Security Centre at WMG, University of Warwick, Coventry, United Kingdom (e-mail: n.sohrabi-safa@warwick.ac.uk).

specific host [6]. In NIDS, the IDS is normally installed before and after the firewall so that traffic for the whole network segment is captured. In the case of HIDS, the IDS focuses on a specific host to examine packets, logs and system calls. As such, HIDS are more suitable for identifying internal attacks compared to NIDS [7].

According to [8], there are two types of IDS: The Signature Based Detection System (SBDS) and the Anomaly Based Detection System (ABDS). In SBDS, a set of previously defined rules are stored in databases and used to identify known attacks. Given that the SBDS technique relies on consistent signature updates, it is unable to detect unknown or new attacks [9]. Consequently, such attacks could pass through the system undetected. On the other hand, the ABDS approach is based on analysis of normal behaviour traffic as a baseline of general usage patterns. Fundamentally, ABDS is based on the assumption that any traffic that deviates from normal patterns can be identified as malicious traffic [10]. The main advantage of this approach is its ability to identify new or unknown attacks. The presented detection approach presented in this research leverages the strength of ABDS.

The ensemble classification technique is a data mining approach that is based on statistical learning theory. It involves a combination of several classifiers to obtain improved performance [11]. The ensemble method is divided into 3 main approaches: (i) bagging, (ii) stack generalisation and (iii) boosting. Bagging, often otherwise referred to as 'bootstrap aggregating', aims to improve detection accuracy by fusing the outputs of learned classifiers into a single prediction. For instance, the Random Forests (RF) algorithm achieves high classification accuracy by fusing random decision trees using the bagging technique. Stack generalisation, or 'stacking', involves the combination of predictions from several learning algorithms. The prediction output from base-level classifiers is used to achieve a high level of generalisation accuracy. The advantage of this algorithm is that it can significantly enhance the generalisation of the learning algorithm and can thereby produce better results than when using single classifiers [12].

In this research, an anomaly based intrusion detection that recognises web attacks run via a HTTP protocol and which uses an ensemble based classification approach is proposed. The Logitboost algorithm is used as a meta-classifier together with Random Forests (RF) as a weak classifier. The performance of the proposed technique is evaluated in the context of intrusion detection. The remainder of this paper is organised as follows. Section 2 reviews the related work on intrusion detection and the proposed approaches are discussed in Section 3. The experimental results are presented in Section 4 while Section 5 concludes and outlines future work.

II. Related Work

In this section, we discuss related work in the areas of feature selection and data mining algorithms used for choosing a classifier for attack detection.

A. Feature Selection

Currently, the two general methods used in feature selection are the filter-based and wrapper-based [13] approaches. Filter-based subset evaluation (FBSE) was introduced to overcome the redundant feature issue that arises when using filter-ranking [14]. It examines the whole subset in a multivariate way, selects relevant features and explores the degree of relationship between them. FBSE is a heuristic-based method that uses probabilities and statistical measures to search for and evaluate the usefulness of all the features that have been identified. Alternatively, wrapper-based subset evaluation (WBSE) uses a classifier to estimate the worth of each feature subset. Usually, WBSE has better predictive accuracy than FBSE. This is because the selection approach is optimised when evaluating each feature subset with a particular classification algorithm. Conversely, most of the time the wrapper-based approach uses a classification algorithm to evaluate each set of features. This has made it excessively expensive to execute. Moreover, when dealing with a large database that consists of many features the wrapper can become uncontrollable [15]. Wrappers are also associated with the classifier's algorithm and that makes it more difficult to shift from one classifier to another since the selection process needs total re-initiation. Unlike wrappers, the selection criteria of filters use distance measures and correlation functions [16]: as such it does not require re-execution for different learning classifiers. The result of this is that its execution is much faster than that of wrapper-based approaches. Filters are best suited to large database environments that contain many features. Researchers have often used the filter, as an alternative to the wrapper, since the latter is expensive and time-consuming to run.

B. Classification

Classification used in IDSs uses a supervised approach that has the capability to differentiate unusual data patterns, and this makes it suitable for identifying unseen attack patterns [17]. A classifier will gather knowledge by training the pre-classified sample that represents classes. It can act as a predictor for some unknown samples or a descriptor for classified samples. Furthermore, classification has been widely used due to its strong performance in identifying normal structure accurately, thereby contributing towards its low rates of false detection [18]. Many previous works, employing method such as Naïve Bayes, Decision Tree, Random Forests, Support Vector Machines (SVM) and Multilayer Perceptron (MLP), have used a single classifier in the field of intrusion detection [17], [19], [20], [21] and [22]. In spite of each classifier having good detection accuracy in detecting specific threats, the processing time varies depending on the complexity of the algorithm used in data processing. This leads to a longer processing time if a large number of instances are involved in building the detection model. This can also lead to higher misclassification rates [23]. Furthermore, certain classifiers take longer processing time in building a detection model. Generally better detection results can be achieved using complex algorithms performing deep analysis of the data instances. For example, MLP can achieve better detection accuracy when compared to SVMor the decision tree algorithm J48 [24], but at the same time, MLP is can be very time-consuming algorithm when compared to RF or the decision-based DTable and J48 [25]. RF is an ensemble approach which consists of many decision

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2766844, IEEE Access

trees (such as J48), and this comes with the advantage that it can process both numerical and categorical data in a way that produces a finer prediction output than J48 alone. Thus, the RF classifier is preferable over the individual J48.

C. *Boosting Algorithms*

Boosting is mainly used to boost a weak classifier or weak learner with the aim of achieving a higher accuracy classifier. In other words, boosting can be considered a meta-learning algorithm. The incorrectly classified instances from the previous model are used to build an ensemble. Weak classifiers such as decision stump, which is based on a decision tree with a root node and two leaf nodes, are often used as a boosting technique [26]. Adaboost (Adaptive boosting) is the most popular boosting algorithm that was first introduced in [27]. The high degree of accuracy which comes along with using this algorithm has attracted researchers to use it in IDS, see [28], [29] and [30].

In [28], the author proposed Adaboost, with a decision stump as a weak classifier. The noise and outliers existing in the dataset are initially removed by training the full data. The sample data that contained high weight is classified as noise and outliers. In that work, although the detection rate achieved was almost 92%, the false alarm rate was still 8.9%. Subsequent research [29] considers choosing the right weak classifier for Adaboost. In the work, the following four classifiers are compared: NNge (Non-Nested generalized examplars); JRip (Extended Repeated Incremental Pruning); RIDOR (Ripple-Down Rule); and Decision Tables as a base classifier for Adaboost. The proposed conjunction of Adaboost with NNge received the highest detection rate in detecting user to root (U2R) and remote to local (R2L) types of attack while a combination of Adaboost with Decision Tables was found to be efficient in detecting Denial of Service (DoS) attacks. More recent work, [30], details a similar concept to [28]. The author tested a naïve Bayes algorithm utilised as a weak classifier. Although the proposed algorithm could achieve a 100% detection rate with respect to DoS attacks, the overall performance (84% detection rate with 4.2% false alarm rate) is still much lower when compared to [28].

The introduction of the Logitboost algorithm [31] was designed as an alternative solution is to address the limitations of Adaboost in handling noise and outliers. The Logitboost algorithm uses a binomial log-likelihood that changes the loss function linearly. In contrast, Adaboost uses an exponential loss function that changes exponentially with the classification error. This is the reason why Logitboost tends to be less sensitive to outliers and noise. To the best of our knowledge, no research to date has investigated the performance of the Logitboost algorithm in the field of ABDS environment.

III. METHODOLOGY

In this research, our anomaly detection approach consists of two parts: pre-processing (hybrid feature selection) and data mining (boosting classification algorithm). Fig. 1 presents the proposed anomaly detection model in detecting web attacks.

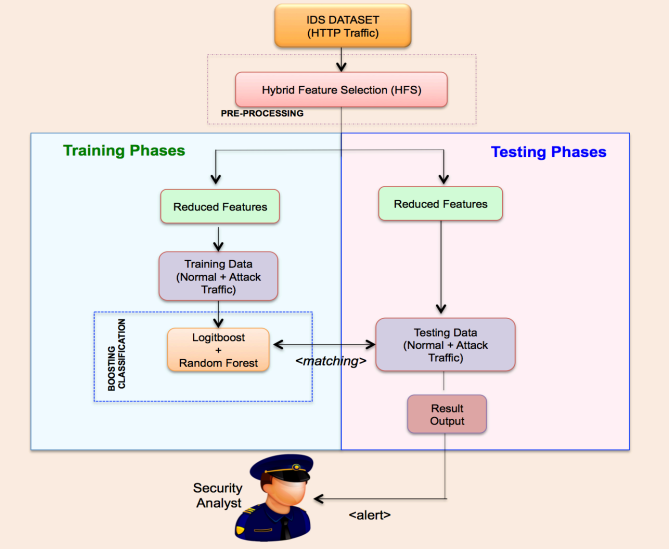

Fig. 1. The Proposed Anomaly Detection Model

A. *Pre-processing*

In the pre-processing step, we adopt the Hybrid Feature Selection (HFS) [32] technique to leverage the strengths of both the filter and wrapper approaches. In addition, the proposed filter-based subset evaluation (FBSE) was utilised to resolve the filter-ranking problem where redundant features were identified.

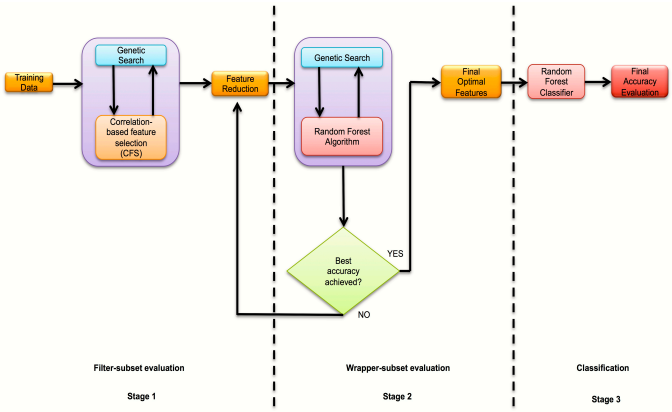

Fig. 2. Hybrid Feature Selection (HFS) design [32]

Fig. 2 illustrates the process flow for building HFS, which can be classified into 3 stages as follows:

In Stage 1, the process starts with the filter-subset evaluation. It processes the original features M and produces a new set L of reduced features, where $L \subseteq M$. We propose the Correlation Feature Selection (CFS) approach for use in Phase 1 due to its robustness in removing redundant and irrelevant features. This approach overcomes the issue of redundant features because in CFS the relationship between features is measured using equation (1). Additionally, the reduced features in feature ranking are usually defined without the need to perform further examination (for example, information gain and gain ratio). The CFS is an intelligible filter algorithm that evaluates subsets of features based on heuristic evaluation functions. The evaluation is based on the hypothesis *"A good feature subset is one that contains*

features highly correlated with the class, yet uncorrelated with each other" [15].

$$Ms = \frac{k\overline{rcf}}{\sqrt{k + k(k-1)\overline{rff}}} \quad (1)$$

Equation (1) shows how the merit function, *M*, is used to select a subset *S* containing *k* number of features. Both redundant and irrelevant features are determined by the $\overline{rcf}$ which represents the mean of the relationship of each feature to its class while $\overline{rff}$ is the mean of the relationship among the features. An exhaustive search is not feasible in large datasets [15] due to the high complexity. As such, we employ heuristic search techniques and chose a genetic algorithm as the search function. This was because our experiment reveals that the genetic algorithm gives a global optimum solution and is more robust than particle swarm optimisation (PSO), best-first and greedy methods. Furthermore, at this stage it is crucial to help to truncate the computational effort using the wrapper approach as it only deals with a reduced set of features compared to the original set of features.

In Stage 2, the reduced feature set *L* gathered from the FBSE was combined with the WBSE method to produce the final set of optimal features *K,* where $K \subseteq L \subseteq M$. The proposed filter and wrapper hybridisation approach leverages the strengths of each to produce a much better result in terms of accuracy, false alarm rate and fewer redundant and irrelevant features. This is due to the fact that the filter approach cannot find the best available subset, since it is less dependent on the classifier. On the other hand, the wrapper approach is proven to be more effective and produces better accuracy. Nevertheless, it is computationally expensive when dealing with a large dataset. Thus, by leveraging the strengths of both methods, we combined them together to form a hybrid feature selection approach. We use the Random Forests (RF) classifier in WBSE to evaluate the selected features using a genetic search and determine the final *K* feature subset. The searches would continue to train a new model for each subset and stop once the final optimum subset is found.

Stage 3 is called the classification stage. In this stage, the final optimum subset *K,* produced by WBSE, was tested by the RF classifier with 10-fold cross validation. RF consists of many decision tree classifiers. Each decision tree was constructed from the different original dataset samples. The outputs were chosen based on votes obtained from each tree that indicated the tree's decision concerning the class object. The most votes for the object are from the best individual trees.

The feature selection procedures were conducted using the training data that contained a mixture of normal and attack traffic. The significance of features are measured using a correlation function in the filter process, while in the wrapper process, a classification algorithm is used. The features that achieve high merit scores and are highly correlated to the class are selected. On the other hand, those features that are highly correlated with other features indicates are redundant, and as such these features are removed in the stages 1 and 2. Further analysis on the features selected by the proposed method are discussed in the next section.

B. Proposed Ensemble Classification Method

In this section, the proposed ensemble classification method based on a boosting algorithm is described. We use the boosting algorithm named *Logitboost* as the meta-classifier for boosting classification. From preliminary experiments and examination of the literature, we found that this algorithm is more suitable for handling noisy and outlier data over the widely used *Adaboost* algorithm. Consider a training data set with *N* samples and divided into two classes (in this study the two classes are abnormal and normal). The two classes are defined as $y \in \{-1,+1\}$, i.e. samples in class y=1 are instances of normal traffic while y=-1 are the samples of abnormal, or attack traffic. Let the set of training data be *{(x1, y1),….., (xi,yi),… (xn,yn)},* where *xi* is the feature vector, and *yn* is the target class. The Logitboost algorithm consists of the following steps [31]:

1) *Input data set N= {(x1, y1),….., (xi, yi),… (xn, yn)}, where xi∈X and yi ∈ Y = {-1,+1}. Input number of iterations K.*
2) Initialized the weights wi= 1/N, i= 1, 2, …, N; start committee function *F(x)*=0 and probabilities estimates P(*xi*)=1/2.
3) *Repeat for k*= 1,2,…, *K*:
   a. Calculate the weights and working response
   $$wi = p(xi)(1 - p(xi)) \quad (2)$$
   $$zi = \frac{yi - p(xi)}{p(xi)(1 - p(xi))} \quad (3)$$
   b. Fit the function *fk* (*x*) by a weighted least squares regression of *z*i to *x*i using weights *w*i. In this research, we use random forests as weak classifier to fit the data using weights *w*i.
   c. Update
   $$F(x) \leftarrow F(x) + \frac{1}{2} fk(x) \quad (4)$$
   and
   $$p(x) \leftarrow \frac{e^{F(x)}}{e^{F(x)} + e^{-F(x)}} \quad (5)$$
4) Output the classifier:
   $$sign\,[F(x)] = sign\left[\sum_{k=1}^{K} fk(x)\right] \quad (6)$$

At this point, *sign* [*F* (*x*)] is a function that has two possible output classes:

$$sign\,[F(x)] = \begin{cases} 1, & if\ F(x) < 0 \\ -1, & if\ F(x) \geq 0 \end{cases} \quad (7)$$

IV. Experimental Results and Analysis

In this section, we present the datasets used for evaluation, the number of samples used in our experiments, experimental tools employed in our proposed approach, and the evaluation metrics adopted to measure the performance of the proposed approach.

A. Dataset Description

The proposed method was analysed through experiments using two different datasets: the NSL-KDD dataset and the UNSW-NB15 dataset. We made use of publicly available

labelled datasets simply to avoid problems with recorded traffic from the real environment, outlined in [33]. Both datasets are available online and have been comprehensively used as a standard benchmark by many researchers in this field [44-53]. The NSL-KDD dataset is the traditional and most commonly used dataset in this field. In essence, the dataset is a modified version of the KDD Cup 1999 dataset, with some redundant traffic removed. In contrast, the UNSW-NB15 dataset is a modern updated dataset, which claims to contain more realistic and modern attack types [34].

The NSL KDD dataset was generated by [35], and based on the KDD 99 dataset generated as part of the DARPA 1998 Intrusion Detection System (IDS) Evaluation dataset project created by Lincoln Lab [36]. The dataset was simulated using artificial data and generated in a closed network, where some of the network involves proprietary network traffic with manual injected attacks.

The simulation is a replication of the medium size traffic found in US Air Force bases in collaboration with Air Force Research Laboratory (AFRL). Since KDD 99 suffered from some drawbacks, the dataset was revised in [35], where removing duplicate and redundant traffic in the dataset was removed. This makes it unsuitable for processing by the learner as learners tend to display a bias towards frequent data. NSL KDD has affected a further improvement by removing 78% and 75% of duplicated traffic in the training and testing data respectively. As presented in Table I, based on the reduced dataset generated, 4,500 and 5,641 instances in the training and testing dataset are from http traffic. There are four categories of class attack as below:

Probe: A probe attack is an attempt to gather or learn specific information in a targeted network or host for reconnaissance reasons (e.g., port scanning). This type of attack is commonly used by an attacker to retrieve information from the machines connected inside the network where the host is vulnerable to this type of attack depending on the type of operating system or version of software installed or used. This kind of attack functions as a preliminary stage for an attacker before they launch an attack which purports to actually compromise the targeted network or host. This class of attack is the extremely common since it requires very little technical skill. Although there is no specific destruction to an organisation caused by these activities, they are still considered a serious threat due to the information obtained by the attacker, which is likely to be useful in launching any future attacks.

Denial of Services (DoS): Denial of Services attacks are class of attack where an attacker sends a huge volume of request connections, normally with the intention of disrupting and paralysing the system server. In short, the attack encompasses destructive characteristics aimed at compromising the targeted network system infrastructure. One example of a DoS attack is when a web service is rendered unable to respond to legitimate users who need access because the server is flooded with innumerable connection requests. DoS attacks are classified based on the degree to which they cause unavailability of service to legitimate users.

User to Root (U2R): The user to root attack is a type of attack during which an attacker exploits the administrative account to gain access to the root in an attempt to retrieve, modify or abuse important resources inside the system. Social engineering is a common method used as part of this attack, involving the attacker gaining access to the victim's account and exploiting a vulnerability in order to gain access as a super user. An example of this kind of attack is buffer overflow, where the attacks is the cause of regular programming errors or system settings mistake.

Remote to User/ Remote to Local (R2L): Remote to user attacks are also known as *remote to local attacks*. This type of attack happens when an attacker exploits a vulnerability in the victim's machine over the network to illegally gain local access as an authorised user. The privilege of this successful attack allows the attacker to gain the status of an authorised user to perform legitimate activities. These common attacks usually involve social engineering. Commonly, the attacker uses a trial-and-error approach by determining the user's password perhaps through some scripting method such as a brute force attack. Some sophisticated approaches involve the attacker successfully installing malicious tools with the intention of capturing the user password before using it to gain access to the system.

The UNSW-NB 15 was simulated by [34] using the IXIA PerfectStorm tool in the Cyber Range Lab at the Australian Centre for Cyber Security (ACCS). The dataset was generated based on the combination of synthetic attack activities along with real modern normal behaviours. Fig. 3 illustrates the test bed configuration of the UNSW-15 dataset. The full dataset contains captured raw traffic of 100GB with nine synthetic types of attacks: Backdoors, DoS, Analysis, Fuzzers, Generic, Worms, Reconnaissance, Shellcode and Exploits.

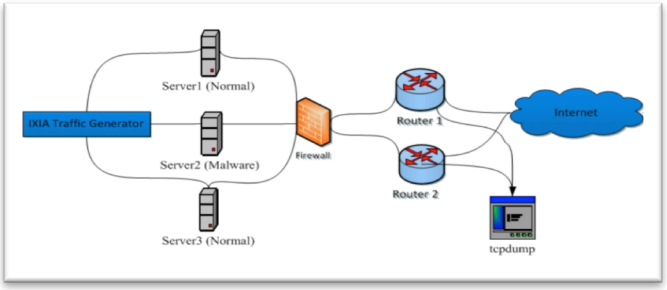

Fig. 3. UNSW-NB15 Testbed Network Architecture [34]

The features and the class labels are generated by Argus and Bro-IDS tools in conjunction with twelve algorithms. The dataset consists of both combination traffic on synthetic attack behaviours and real normal activities. The total traffic captured is 2,540,044 where parts of this data are divided into two sets (training and testing), which consisting of 175,341 and 82,332 instances of traffic respectively. In these sets of traffic, 8,287 instances in the training data and 18,724 instances in the testing data are based on http traffic as presented in Table II.

TABLE I: DISTRIBUTION OF HTTP TRAFFIC FOR NSL-KDD DATASET

NSL-KDD Dataset	Training Data		Testing Data	
	Normal	Attack	Normal	Attack
	3,817	683	2,856	2,785

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2766844, IEEE Access

TABLE II: Distribution of HTTP Traffic for UNSW-NB 15 Dataset

UNSW-NB 15 Dataset	Training Data		Testing Data	
	Normal	Attack	Normal	Attack
	4,013	4,274	5,348	13,376

B. *Experimental Settings*

The detection performance of the proposed ensemble approach applied to both the NSL-KDD and the UNSW-NB15 datasets are presented in this section. The experiments were conducted on a 2.4 GHz Pentium Core i7 with 8GB RAM running the Windows 7 operating system.

In this study, the results are obtained using the default setting of Weka data mining and machine learning software (version 3.8) [37] along with MySQL for the database management system. Weka is open source software written in Java and developed at University of Waikato, New Zealand. It comprises many machine learning and data mining techniques used for knowledge discovery.

C. *IDS evaluation method*

There are number of performance metrics that can be used to evaluate the performance of an IDS. The most commonly used metrics in the field of intrusion detection are focused on the false alarm rate (FAR), detection rate (DR) and accuracy (ACC) [38]. In this research, we had employed the same performance metrics evaluate our proposed methods:

a) False Alarm Rate (FAR): This quantifies the amount of benign traffic detected as malicious traffic
b) Detection Rate (DR): This quantifies the proportion of detected attacks among all attack data
c) Accuracy (ACC): This measures in percentage form, where instances are correctly predicted

$$\text{False Alarm Rate (FAR)} = \frac{\text{FP}}{\text{FP} + \text{TN}} \quad (8)$$

$$\text{Detection Rate (DR)} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (9)$$

$$\text{Accuracy (ACC)} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (10)$$

D. *Pre-processing of the NSL KDD and the UNSW-NB15 datasets*

In the pre-processing phase, we adopted the HFS approach for both datasets to select the most prominent features. As presented in Tables III and IV, the original 41 NSL-KDD and the original 43 UNSW-NB15 features were reduced to 10 and 5 respectively. This significant reduction of features has contributed to reducing the overall detection time during the experiment.

As can be seen from Fig. 4 and 5, the reducing features through the use of HFS allow us to obtain slightly better performance in term of false positive rate, detection rate and accuracy rate over the original full features for both NSL KDD and UNSW-NB 15 datasets. This indicates that the HFS technique is suitable for removing irrelevant and redundant features residing in the datasets.

TABLE III: Feature selection for NSL-KDD dataset

Feature Selection Approach	Number of Features	Feature Selection
Original Features	41	*f1, f2, f3, f4, f5, f6, f7, f8, f9, f10, f11, f12, f13, f14, f15, f16, f17, f18, f19, f20, f21, f22 f23, f24, f25, f26, f27, f28, f29, f30, f31, f32, f33, f34, f35, f36, f37, f38, f39, f40, f41*
Salient Features (Hybrid Feature Selection)	10	*f5, f23, f24, f29, f31, f33, f34, f35, f37, f39*

TABLE IV: Feature selection for UNSW-NB 15 dataset

Feature Selection Approach	Number of Features	Feature Selection
Original Features	43	*f1, f2, f3, f4, f5, f6, f7, f8, f9, f10, f11, f12, f13, f14, f15, f16, f17, f18, f19, f20, f21, f22 f23, f24, f25, f26, f27, f28, f29, f30, f31, f32, f33, f34, f35, f36, f37, f38, f39, f40, f41, f42, f43*
Salient Features (Hybrid Feature Selection)	5	*f8, f25, f26, f29, f31*

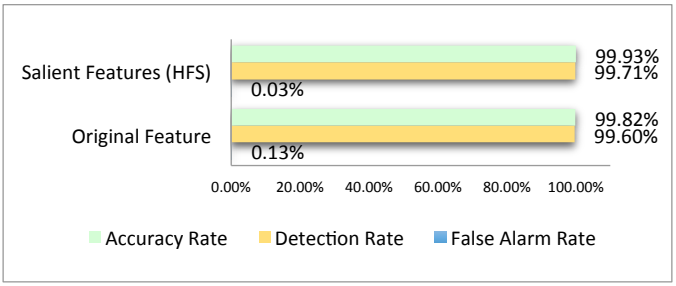

Fig. 4. Comparison of performance between original and reduced features in the NSL-KDD dataset

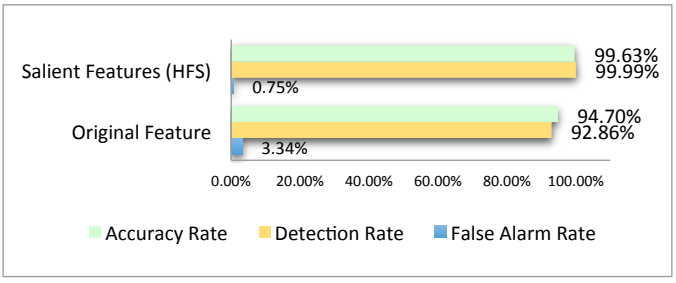

Fig. 5. Comparison of performance between full and reduced features in the UNSW-NB15 dataset

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2766844, IEEE Access

The final sets of features selected by the HFS process are the features that contributed most significantly to determining traffic behaviours. For instances in the NSL KDD dataset, DoS and Probe attacks are triggers when there are many connections involving to the same hosts. The significant features for observing these types of attacks are src_bytes, count and srv_count. Meanwhile, in an R2L attack, the attacker behaviour in accessing the local account mean that features such as src_bytes, same_srv_rate and dest_host_srv_count become relevant for attack detection. On the other hand, in UNSW NB-15, the TCP connection type round trip time *"tcprtt"* and SYN time between traffic "*synack*" features are significant in the identification of DoS attacks. Meanwhile the feature of response_body_len is important for recognising Reconnaissance attacks. The 10 features selected by HFS in NSL KDD are: *src_bytes, count,srv_count,same_srv_rate,srv_diff_host_rate,dst_host_srv_count,dst_host_same_srv_rate,dst_host_diff_srv_rate,dst_host_srv_diff_host_rate and dst_host_srv_serror_rate.* Further, the 5 features selected by HFS in UNSW-NB 15 are: *sbytes, tcprtt, synack, dmean and response_body_len.* In terms of significant features that contribute to recognising attack behaviour, our proposed HFS approach has successfully selected all significant features that are relevant to the attack types.

TABLE V: Detection result derived by proposed approach for NSL-KDD testing dataset

Attack Category	Attack Name	Attack Traffic in the Training Dataset	Attack Traffic in the Testing Dataset	Attack Traffic Detected by Proposed Approach	%age of Detected Attack
DOS	back	203	1112	**1112**	**99.75**
	apache2	434	302	**301**	
	neptune	44	1334	**1328**	
PROBE	portsweep	1	16	**11**	**54.83**
	ipsweep	-	7	**0**	
	satan	-	7	**6**	
	nmap	-	1	**0**	
	saint	1	-	**-**	
R2L	phf	-	6	**1**	**16.67**
Total	-	682	2,785	**2,759**	**100**

While the anomaly detection utilises data mining approaches, Logitboost classification is employed for detecting known and unknown attack traffic. For Logitboost combination, we choose RF as a base learner due to its robustness in dealing with noise and outliers data and its superior performance over other classifiers.

To test the robustness of our proposed approach, we have ensured that the attack traffic in both training and testing data were significantly different. In simple terms, this means that the sample attack traffic used in the training data is not itself part of the testing data. In addition, we made sure that the proportion of attack traffic in the testing data was higher than the attack traffic in the training dataset. For example, in this research 2,785 and 13,376 instances of attack traffic used in the testing data were available for detection, whilst 683 and 4,274 instances of attack traffic were used to build the classification model in the NSL-KDD and UNSW-NB15 training sets respectively.

As presented in Table V, there are, in total, 9 types of attack in the NSL-KDD dataset. In the training dataset, there are 5 types of attack present: *back, apache2, neptune, portsweep and saint* whilst 8 types of attack: *back, apache2, neptune, portsweep, ipsweep, satan, nmap and phf* are in the testing dataset. It can be seen that 4 out of 8 types of attack "*ipsweep, satan, nmap and phf"* in the testing dataset are new attacks, which are not available in the training dataset. Among all the attack traffic present in the testing data, our proposed ensemble approach successfully recognised 99.10% instances of attack traffic. The attack type with the highest detection rate is DoS with 99.75%, followed by Probe with 54.83% and the lowest is the R2L with 16.67%. As a result of further investigation, the poor performance of R2L was determined to be due to the feature named "dest_host_diff_srv_rate", which contains similar values with the feature of normal traffic, thus the classifier is overly keen to recognise R2L attacks as normal traffic. In addition, since the connection of R2L and normal is similar, it is almost impossible for the system to distinguish between these two classes.

Data set	Back door	Fuzzers	Reconnaissance	Exploits	Analysis	DoS	Worms	Generic
Training instances	9	251	470	2804	-	493	34	213
Testing instances	83	836	1603	8677	558	1216	114	289
Detection Rate (%)	100	80.98	98.75	94.33	6.63	87.5	99.12	91

TABLE VI: Detection result derived by proposed approach for UNSW-NB15 testing set

Additionally, R2L attacks occur when an attacker has tried to gain unauthorised access to a local machine, thus the relevant kind of traffic seems legitimate which makes it difficult for the system to recognise it as an attack. Meanwhile for the Probe attack type, the main cause of the low detection percentage is brought about by the three new attack types which only exist in the testing dataset. Although we have not been able to recognise all three new attack types, it is worth

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2766844, IEEE Access

mentioning that we successfully recognised one of the new attack types, (*satan)* in 85.71% of occurrences clearly indicating that our proposed ensemble approach is capable of detecting unknown attacks with good performance.

Table VI presents the attack traffic tabulated for both the training and testing datasets along with the results obtained using our approach on the UNSW-NB15 dataset. As mentioned in [Section IV, A] this dataset is comprised of real normal traffic combined with a variety of imbalanced synthetic attack traffic, which results in this dataset being more challenging to evaluate. In the training dataset, there are 7 types of attack present: *backdoor, fuzzers, reconnaissance, exploits, dos, worms and generic* whilst 8 types of attack: *backdoor, fuzzers, reconnaissance, exploits, analysis, DoS, worms and generic* are in the testing dataset. As can be seen, the main difference between the testing data and the training data is that it contains a new attack type named *"analysis"*. Our proposed ensemble approach successfully obtained an 89.75% detection rate among all attack traffic existing in the testing dataset.

The attack type with the highest detection rate is *backdoor* with 100% detection, followed by *worms* with 99.12%, *reconnaissance* with 98.75%, *exploits* with 94.33%, *generic* with 91%, *dos* with 87.5%, *fuzzers* with 80.98% and the lowest is *analysis* with 6.63%. The results show that with respect to five out of eight types of attack, our approach achieved a detection rate of more than 90%. The low detection rate of "analysis" is due to the unavailability of samples residing in the training dataset, which make it difficult for the system to classify it as an attack. In spite of achieving the lowest detection rate, the system is still able to recognise *"analysis"* 6.63% of the time.

TABLE VII: Performance comparisons obtained by the proposed method and other previous work on KDD and NSL-KDD as reported in [48]

Methods	Feature Selection	Features	Normal (%)	DoS (%)	Probe (%)	R2L (%)	U2R (%)	Detection Rate (%)	False Alarm Rate (%)
ACC [39]	Yes	N/A	98.8	97.3	87.5	12.6	30.7	N/A	N/A
GP Transformation Function [40]	No	41	**99.93**	98.81	**97.29**	45.2	80.22	N/A	N/A
Hierarchical SOM [41]	No	41	98.4	96.9	67.6	7.3	15.7	90.6	1.57
MOGFIDS [42]	Yes	25	98.36	97.20	88.59	15.78	11.01	92.76	N/A
Multinominal Naïve Bayes [43]	No	41	N/A	N/A	N/A	N/A	N/A	96.5	3.0
GHSOM-MOF [44]	Yes	29	N/A	N/A	N/A	N/A	N/A	99.12	2.24
N-KPCA-SVM [45]	Yes	N/A	N/A	N/A	N/A	N/A	N/A	95.26	1.03
OS-LEM [46]	Yes	21	99.07	99.14	90.35	78.10	56.75	97.67	1.74
TVCPSO-SVM [47]	Yes	17	99.13	98.84	89.29	67.84	40.38	97.03	0.87
Ramp-KSVCR [48]	No	41	99.14	99.49	93.58	**91.09**	**68.75**	98.48	0.86
Proposed Method	Yes	10	99.82	**99.75**	54.83	16.67	N/A	**99.10**	**0.18**

TABLE VIII: Performance comparisons obtained by the proposed method with other approaches on UNSW-NB15 as reported in [34].

Classifiers	Accuracy rate %age	False alarm rate %age
DT	85.56	15.78
LR	83.15	18.48
NB	82.07	18.56
ANN	81.34	21.13
EM clustering	78.47	23.79
Proposed Method	**90.33**	**8.22**

Tables VII and VIII show the performance of our proposed method in terms of *FAR*, *DR* and *ACC* and compared to the previous methods tested on the KDD and NSL-KDD datasets as reported in [48]. For the UNSW-NB15 dataset, it should be noted that since this dataset is recently published, there are a limited number of research experiments conducted using it. As such, the obtained results derived from our method compare favourably with five techniques including: DT, LR, NB, ANN and EM clustering as published in [34]. The best results are highlighted in boldface font. It needs to be mentioned that the reported result comparisons are for reference only due to the fact that comprehensive resemblance is not an easy task as different researchers have used different proportions of traffic types, sampling methods, computational time and pre-processing methods. In most cases, the comprehensive comparison becomes more difficult since these details are not provided. As presented in Tables VII and VIII, although our proposed approach achieves better performances in most of the cases, it cannot be claimed that the proposed method outperformed others in terms of performance. Nevertheless, our proposed approach does show some ability with a robust performance in detecting unknown attack traffic, which did not exist in the training dataset.

TABLE X: Comparison of FAR, DR and ACC with other six algorithms in UNSW-NB15 dataset

Algorithms	Model Build (sec)	Detection Time (sec)	False Alarm Rate (%)	Detection Rate (%)	Accuracy (%)
Naïve Bayes (NB)	0.02	0.11	19.18	42.73	53.61
Support Vector Machine (SVM)	3.89	7.37	32.55	87.00	87.41
Multi Layer Perceptron (MLP)	2.60	0.08	6.50	53.43	64.86
Decision Tree (J48)	0.06	0.04	6.68	88.23	89.68
Random Forests (RF)	0.14	0.06	7.89	89.32	90.11
Adaboost + Random Forests (RF)	1.66	0.74	8.30	89.71	90.27
Logitboost + Random Forests (RF)	**1.72**	**0.65**	**8.22**	**89.75**	**90.33**

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2766844, IEEE Access

TABLE IX: Comparison of FAR, DR and ACC with other six algorithms in NSL-KDD dataset

Algorithms	Model Build (s)	Test Time (s)	False Alarm Rate (%)	Detection Rate (%)	Accuracy (%)
Naïve Bayes (NB)	0.02	0.06	1.54	98.20	98.33
Support Vector Machine (SVM)	0.21	0.23	0.21	95.87	97.86
Multi Layer Perceptron (MLP)	3.42	0.05	0.42	97.92	98.76
Decision Tree (J48)	0.06	0.04	0.21	98.24	99.03
Random Forests (RF)	0.16	0.08	0.21	98.38	99.10
Adaboost + Random Forests (RF)	0.27	0.17	0.18	98.64	99.24
Logitboost + Random Forests (RF)	**0.35**	**0.21**	**0.18**	**99.10**	**99.45**

In addition, it should be noted that we evaluated the performance of the proposed approach with some eminent state-of-the-art data mining algorithms used in IDSs. Tables IX and X display a comparison of performance metrics between our proposed approach and six other data mining algorithms previously used by researchers in IDSs: Naïve Bayes [17], Support Vector Machine [19], Multilayer Perceptron [20], Decision Tree [21], Random Forests [22] and Adaboost [28]. Five single classifiers are evaluated individually in terms of the time taken to build classification models, detection time, false alarm rate, detection rate and accuracy rate to choose a better combination for the Logitboost classifier as shown in Tables IX and X. This is a crucial aspect of our research because the algorithm choice needs to be further re-classified with ensemble approaches for better detection performance.

In the NSL-KDD dataset, RF had shown comparable performance in terms of the accuracy, detection rate and false alarm rate. Although J48 had shown a faster detection time by 50% over RF, the detection and accuracy rate achieved by RF is slightly better than J48. Meanwhile, in the UNSW-NB15 dataset, RF outperformed every single other classifier by achieving 90.11% detection accuracy. Thus, in our proposed detection approach, we chose RF as a base classifier to ensemble with the Logitboost classifier for both the NSL-KDD and UNSW-NB15 datasets.

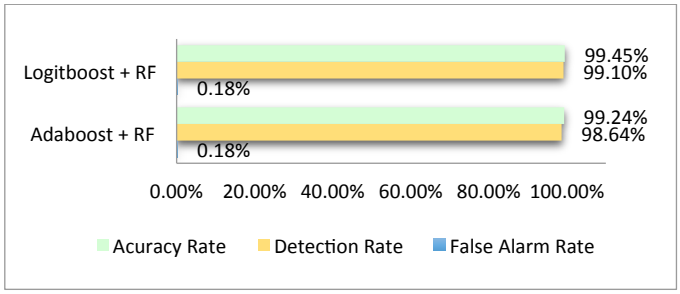

Fig. 6. Comparison of performance algorithms in NSL-KDD dataset

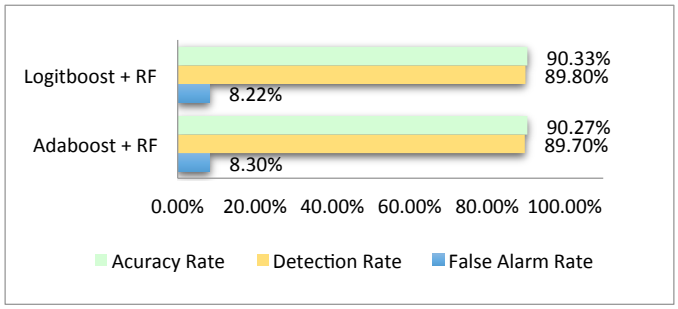

Fig. 7. Comparison of performance algorithms in UNSW-NB15 dataset

Additionally, a further experiment is performed for fair comparison of our proposed boosting algorithm with the Adaboost boosting algorithm. As presented in Fig. 6 and 7, our proposed approach shows slightly better performance in terms of detection rate and accuracy rate over Adaboost with 0.46% and 0.1% and with 0.21% and 0.06% for NSL-KDD and UNSW-NB15 datasets respectively. From the aforementioned results, we conclude that our algorithm provides a comparable detection accuracy rate with a low false alarm rate, which is the most crucial property of IDSs in practice.

V. Conclusion and Future Work

There have been numerous anomaly intrusion detection studies conducted in the past. Nevertheless, achieving exceptionally low false alarm rates with high attack recognition capabilities for unseen attacks remains a major challenge. In this paper we have presented the Logitboost-based classifier for detecting known and unknown web attack traffic. The proposed approach was evaluated using two publicly available labelled intrusion detection evaluation datasets NSL-KDD and UNSW-NB15 to allow different integration testing environments. In pre-processing, redundant and irrelevant features were filtered-out to obtain the most prominent features. Following that, we employed a data mining approach using the Logitboost classifier algorithm to achieve high detection accuracy while preserving a low false alarm rate. The experimental results have demonstrated that our proposed ensemble approach has successfully recognised some unknown attacks and achieved comparable performance with other established state-of-the-art IDS algorithms. Moving forward, the final successful results will be transformed into signatures and stored inside a database. By doing this, detection time can be drastically reduced, since the new entry traffic can be matched with benign/malicious signatures generated from previous detection. Finally, the proposed ensemble approach can be evaluated online using larger, as well as the latest, encrypted traffic.

References

[1] W. Koff and P. Gustafson, "CSC LEADING EDGE FORUM Data rEvolution," *CSC LEADINGedgeforum*, p. 68, 2011.

[2] Cyber Attacks, "Cyber Attacks," 2016. [Online]. Available: http://www.reuters.com/article/us-usa-cyber-idUSKCN12L1ME. [Accessed: 21-Oct-2016].

[3] S. Jones, "NHS seeks to recover from global cyber-attack as security concerns resurface," 2017. [Online]. Available: https://www.theguardian.com/society/2017/may/12/hospitals-across-england-hit-by-large-scale-cyber-attack. [Accessed: 02-Jun-2017].

[4] S. V. Thakare and D. V. Gore, "Comparative Study of CIA," *2014 Fourth Int. Conf. Commun. Syst. Netw. Technol.*, pp. 713–718, 2014.

[5] W. Bul'ajoul, A. James, and M. Pannu, "Improving network intrusion detection system performance through quality of service configuration and parallel technology," *J. Comput. Syst. Sci.*, vol. 81, no. 6, pp. 981–999, 2015.

[6] C. M. Ou, "Host-based intrusion detection systems adapted from agent-based artificial immune systems," *Neurocomputing*, vol. 88, pp. 78–86, 2012.

[7] K. L. I. Iii, "Anomaly Detection for HTTP Intrusion Detection: Algorithm Comparisons and the Effect of Generalization on Accuracy," no. May, p. 196, 2007.

[8] C.-M. Chen, Y.-L. Chen, and H.-C. Lin, "An efficient network intrusion detection," *Comput. Commun.*, vol. 33, no. 4, pp. 477–484, 2010.

[9] P. Louvieris, N. Clewley, and X. Liu, "Effects-based feature identification for network intrusion detection," *J. Neurocomputing*, 2013.

[10] M. Grill, T. Pevny, and M. Rehak, "Reducing false positives of network anomaly detection by local adaptive multivariate smoothing," *J. Comput. Syst. Sci.*, vol. 83, no. 1, pp. 43–57, 2017.

[11] G. Folino and P. Sabatino, "Ensemble based collaborative and distributed intrusion detection systems: A survey," *J. Netw. Comput. Appl.*, vol. 66, pp. 1–16, 2016.

[12] Y. Ren, "An Integrated Intrusion Detection System by Combining SVM with AdaBoost," *J. Softw. Eng. Appl.*, vol. 7, no. November, pp. 1031–1038, 2014.

[13] F. Amiri, M. Rezaei Yousefi, C. Lucas, A. Shakery, and N. Yazdani, "Mutual information-based feature selection for intrusion detection systems," *J. Netw. Comput. Appl.*, vol. 34, no. 4, pp. 1184–1199, 2011.

[14] S. Solorio-Fernandez, J. A. Carrasco-Ochoa, and J. F. Martinez-Trinidad, "A new hybrid filter-wrapper feature selection method for clustering based on ranking," *Neurocomputing*, vol. 214, pp. 866–880, 2016.

[15] M. A. Hall, "Correlation-based Feature Subset Selection for Machine Learning," Hamilton, New Zeland, 1999.

[16] N. Cleetus, "Genetic algorithm with Different Feature Selection Method for Intrusion Detection," *First Int. Conf. Comput. Syst. Commun.*, no. December, pp. 220–225, 2014.

[17] D. M. Farid, L. Zhang, C. M. Rahman, M. A. Hossain, and R. Strachan, "Hybrid decision tree and naïve Bayes classifiers for multi-class classification tasks," *Expert Syst. Appl.*, vol. 41, no. 4 PART 2, pp. 1937–1946, 2014.

[18] Z. Muda, W. Yassin, M. N. Sulaiman, and N. I. Udzir, "Intrusion Detection based on K-Means Clustering and Naïve Bayes Classification," *7th Int. Conf. IT Asia Intrusion*, pp. 1–6, 2011.

[19] S. Zaman and F. Karray, "Features Selection for Intrusion Detection Systems Based on Support Vector Machines," *2009 6th IEEE Consum. Commun. Netw. Conf.*, pp. 1–8, 2009.

[20] D. Parikh and T. Chen, "Data fusion and cost minimization for intrusion detection," *IEEE Trans. Inf. Forensics Secur.*, vol. 3, no. 3, pp. 381–389, 2008.

[21] K. a Jalil, M. H. Kamarudin, and M. N. Masrek, "Comparison of Machine Learning algorithms performance in detecting network intrusion," *Netw. Inf. Technol. ICNIT 2010 Int. Conf.*, pp. 221–226, 2010.

[22] A. Jain, Bhupendra, and R. J.L., "CLASSIFIER SELECTION MODELS FOR INTRUSION DETECTION SYSTEM (IDS)," *Informatics Eng. an Int. J. (IEIJ),* vol. 4, no. 1, pp. 1–11, 2016.

[23] M. Panda, A. Abraham, and M. R. Patra, "A hybrid intelligent approach for network intrusion detection," *Procedia Eng.*, vol. 30, no. 2011, pp. 1–9, 2012.

[24] P. A. Raj Kumar and S. Selvakumar, "Distributed denial of service attack detection using an ensemble of neural classifier," *Comput. Commun.*, vol. 34, no. 11, pp. 1328–1341, 2011.

[25] H. Tribak, B. L. Delgado-Marquez, P. Rojas, O. Valenzuela, H. Pomares, and I. Rojas, "Statistical analysis of different artificial intelligent techniques applied to Intrusion Detection System," *2012 Int. Conf. Multimed. Comput. Syst.*, pp. 434–440, 2012.

[26] S. Fakhraei, H. Soltanian-Zadeh, and F. Fotouhi, "Bias and stability of single variable classifiers for feature ranking and selection," *Expert Syst. Appl.*, vol. 41, no. 15, pp. 6945–6958, 2014.

[27] Y. Freund and R. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," *Comput. Learn. theory*, vol. 55, pp. 119–139, 1995.

[28] W. Hu, W. Hu, and S. Maybank, "AdaBoost-based algorithm for network intrusion detection," *IEEE Trans. Syst. Man. Cybern.*, vol. 38, no. 2, pp. 577–83, 2008.

[29] M. Panda and M. R. Patra, "Ensembling rule based classifiers for detecting network intrusions," *ARTCom 2009 - Int. Conf. Adv. Recent Technol. Commun. Comput.*, pp. 19–22, 2009.

[30] W. Li and Q. Li, "Using Naive Bayes with AdaBoost to Enhance Network Anomaly Intrusion Detection," *Intelligent Networks and Intelligent Systems (ICINIS), 2010 3rd International Conference on*. pp. 486–489, 2010.

[31] J. Friedman, T. Hastie, and R. Tibshirani, "Additive Logistic Regression," *The Annals of Statistics*, vol. 28, no. 2. pp. 337–374, 2000.

[32] Kamarudin, M.H., Maple, C. and Watson, T. (in press) "Hybrid feature selection technique for intrusion detection system," *Int. J. High Perform. Comput. Netw*

[33] P. Mell, V. Hu, R. Lippmann, J. Haines, and M. Zissman, "An overview of issues in testing intrusion

detection systems," *Tech. Rep. NIST IR 7007, Natl. Inst. Stand. Technol.*, pp. 1–21, 2003.

[34] N. Moustafa and J. Slay, "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Inf. Secur. J. A Glob. Perspect.*, vol. 25, no. 1–3, pp. 18–31, 2016.

[35] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," *IEEE Symp. Comput. Intell. Secur. Def. Appl. CISDA 2009*, no. Cisda, pp. 1–6, 2009.

[36] R. Lippmann and J. Haines, "Analysis and Results of the 1999 DARPA Off-Line Intrusion Detection Evaluation," *Recent Adv. Intrusion Detect. Third Int. Work.*, pp. 16–182, 2000.

[37] WEKA, "Machine Learning 'WEKA.'" [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka. [Accessed: 05-Jan-2017].

[38] W. C. Lin, S. W. Ke, and C. F. Tsai, "CANN: An intrusion detection system based on combining cluster centers and nearest neighbors," *Knowledge-Based Syst.*, vol. 78, no. 1, pp. 13–21, 2015.

[39] C. H. Tsang and S. Kwong, "Ant colony clustering and feature extraction for anomaly intrusion detection," *Stud. Comput. Intell.*, vol. 34, no. 2006, pp. 101–123, 2006.

[40] K. M. Faraoun and A. Boukelif, "Multi-Category Pattern Classification Applied," *Int. J. Comput. Intell. Appl.*, vol. 6, no. 1, pp. 77–99, 2006.

[41] H. Gunes Kayacik, A. Nur Zincir-Heywood, and M. I. Heywood, "A hierarchical SOM-based intrusion detection system," *Eng. Appl. Artif. Intell.*, vol. 20, no. 4, pp. 439–451, 2007.

[42] C. H. Tsang, S. Kwong, and H. Wang, "Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection," *Pattern Recognit.*, vol. 40, no. 9, pp. 2373–2391, 2007.

[43] M. Panda, A. Abraham, and M. R. Patra, "Discriminative multinomial naive bayes for network intrusion detection," *2010 6th Int. Conf. Inf. Assur. Secur. IAS 2010*, pp. 5–10, 2010.

[44] E. De La Hoz, E. De La Hoz, A. Ortiz, J. Ortega, and A. Martinez-Alvarez, "Feature selection by multi-objective optimisation: Application to network anomaly detection by hierarchical self-organising maps," *Knowledge-Based Syst.*, vol. 71, pp. 322–338, 2014.

[45] F. Kuang, W. Xu, and S. Zhang, "A novel hybrid KPCA and SVM with GA model for intrusion detection," *Appl. Soft Comput. J.*, vol. 18, pp. 178–184, 2014.

[46] R. Singh, H. Kumar, and R. K. Singla, "An intrusion detection system using network traffic profiling and online sequential extreme learning machine," *Expert Syst. Appl.*, vol. 42, no. 22, pp. 8609–8624, 2015.

[47] S. M. Hosseini Bamakan, H. Wang, T. Yingjie, and Y. Shi, "An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos particle swarm optimization," *Neurocomputing*, vol. 199, pp. 90–102, 2016.

[48] S. M. Hosseini Bamakan, H. Wang, and Y. Shi, "Ramp loss K-Support Vector Classification-Regression; a robust and sparse multi-class approach to the intrusion detection problem," *Knowledge-Based Syst.*, vol. 126, pp. 113–126, 2017.

Muhammad Hilmi Kamarudin obtained his BSc in Computer Network from the Universiti Putra Malaysia, Selangor, Malaysia in 2007, and MSc in Computer Network from the Universiti Teknologi Mara, Selangor, Malaysia in 2010. He is currently pursuing his PhD in Network Security with the University of Warwick, Coventry, UK. His six-years in industrial experience includes operation and maintenance of multi-vendor network security equipment's. His research interests include network security, digital forensics, cyber crime and data mining.

Carsten Maple is Professors of Cyber Systems Engineering and Director of Cyber Security Research in the Cyber Security Centre, WMG at the University of Warwick. He has published over 200 peer-reviewed papers and is the technical lead for the Privacy and Trust Theme in the UK Research Hub for Cyber Security of the Internet of Things, PETRAS, a £23M EPSRC-funded project and is leads the GCHQ-EPSRC Academic Centre of Excellence in Cyber Security Research at the University of Warwick. He is a Faculty Fellow of the Alan Turning Institute and the Chair of the Council of Professor and Heads of Computing in the UK. His research interests are in the security and resilience of cyber-physical systems, authentication, attribution and privacy.

Tim Watson is the Director of the WMG Cyber Security Centre at the University of Warwick. With more than twenty-five year's experience working with industry and in academia, he has been involved with a wide range of systems on several high-profile projects and has acted as a consultant for some of the largest telecoms, power and transport companies. He is an adviser to various parts of the UK government and to several professional and standards bodies. Tim's recent research includes EU-funded projects on combating cyber crime, UK MoD research into automated defence, insider threat and secure remote working, and EPSRC-funded research, focusing on the protection of critical national infrastructure against cyber attack. He is a regular keynote speaker and media commentator on cyber security.

Nader Sohrabi Safa successfully finished his postdoctoral at the Centre for Research in Information and Cyber Security (CRICS), School of ICT, Nelson Mandela Metropolitan University, Port Elizabeth, South Africa. He received his PhD from Faculty of Computer Science and Information Technology, Information System Department, University of Malaya. He is a member of IFIP TC 11 Working Group 12. He also is a member of committee in several annual conferences and reviewer in several journals. His research focuses on Human Aspects of Information Security in Organizations in postdoctoral study. The findings of his studies have been published in prominent journals in this domain. He concurrently works in two important projects: 1) Internet of Things (IoT) in Uk (PETRAS), impact committee. 2) Protecting data in industry. He also collaborates in Global Research Priority (GRP) in University of Warwick.